CNNA-94 Third IEEE International Workshop on Cellular Neural Networks and their Applications
Rome, Italy, December 18-21,1994

# Weight-Control Strategy for Programmable CNN Chips

S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez and R. Carmona
Centro Nacional de Microelectrónica-Universidad de Sevilla
Edificio CICA, C/Tarfia s/n, 41012-Sevilla, SPAIN
Phone: 34 - 5 - 423 99 23. Fax: 34 - 5 - 462 45 06. E-mail: espejo@cnm.us.es

***Abstract*** *- This paper describes a hybrid weight-control strategy for the VLSI realization of programmable CNNs, based on automatic adaptation of analog control signals to levels specified by digital words. This approach merges the advantages of digital and analog programmability, achieving low areas and reduced number of control lines, simplifying the control and storage of the weight values, and eliminating their dependency on global process-parameter variations.*

## 1. Introduction

The implementation of general-purpose programmable CNN systems is a requisite for the application of this computation paradigm in many areas of great interest [1]. In the design of this class of systems, one of the major trends is the optimization (in terms of area and power efficiency, accuracy, and speed) of the programmable scaling blocks or "multipliers". The design of the programmable scaling block is intrinsically related to the type of programmability selected: *analog* or *digital*. This contribution analyzes the two possibilities, and proposes a hybrid (analog/digital) approach which combines the advantages of the two alternatives and virtually eliminates their drawbacks.

## 2. Programmable scaling blocks for CNNs

The programmable scaling blocks required for programmable CNN implementations can be represented as in Fig. 1. Both $x_i$ and $W$ are input signals, while $x_o$ is an output. We refer to $x_i$ as the input signal or simply *the input*, while $W$ is called the *weight signal*. The input is driven by the signal to be scaled (the output variable), which is a function of time during network operation. On the other hand, the weight signal is time-invariant during the CNN process. The ideal behavior of the scaling block can be formulated as follows,

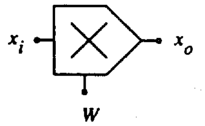

***Figure 1** - Representation of a programmable CNN "multiplier"*

$$x_o = P(W) \cdot S(x_i) \tag{1}$$

where $S(.)$ is a linear and continuous function of $x_i$ in some range around $x_i = 0$. Function $S(.)$ is normalized to the value of its derivative at $x_i = 0$

$$\left.\frac{dS}{dx_i}\right|_{x_i = 0} = 1 \tag{2}$$

and hence, $P(W)$ represents a scaling factor within the linear range of $S(.)$, in which $S(x_i) = x_i$. In general $P(.)$ is not required to be linear, and it can be either a continuous or discrete function. Note that the functionality of the scaling block is not exactly that of a linear analog multiplier.

0-7803-2070-0/94/$4.00 © 1994 IEEE

We will be however use the term *multiplier* for simplicity.

Two general classes of multipliers can be considered attending to the nature of the weight signal $W$. *Digitally-programmed* multipliers are driven by a digital weight signal, and hence, function $P(W)$ needs to be defined only for a discrete set of values of its variable. *Analog-programmed* multipliers are driven by an analog weight signal, and hence, function $P(W)$ needs to be defined within a continuous set of values of its variable. The use of multipliers of either class presents important advantages and drawbacks for CNN implementations.

## 3. Digitally-programmed multipliers

The weight signal of a digitally-programmed multiplier is commonly represented by a binary number of several bits. We assume here a particular representation with N+1 bits, the first of them indicating the sign of the signal, and the remaining N bits indicating the absolute value of the weight. That is,

$$W \equiv (w_s, w_0, w_1, \dots, w_{N-1}) \tag{3}$$

where $w_s$ and $w_0, w_1, \dots, w_{N-1}$ are either "1" or "0", and

$$\mathrm{sgn}(W) = \{ \begin{matrix} -1, \text{if } w_s = 1 \\ 1, \text{if } w_s = 0 \end{matrix}$$
$$|W| = \sum_{i=0}^{N-1} w_i 2^i \le 2^N - 1 \tag{4}$$

which can be summarized by

$$W = (1 - 2w_s) \sum_{i=0}^{N-1} w_i 2^i \tag{5}$$

Note from Eq. 4 that the maximum absolute value of $W$ is $2^N-1$. In order to achieve a weight range of $[-P_{\max}, P_{\max}]$, we associate the following weight increment to each unitary increment in $W$,

$$\Delta P = P_{\max} / (2^N - 1) \tag{6}$$

and define the weight function $P(.)$ as

$$P(W) = \Delta P \cdot W = \frac{P_{\max}}{2^N - 1} (1 - 2w_s) \sum_{i=0}^{N-1} w_i 2^i \tag{7}$$

An schematic implementation of a digital multiplier using this weight codification is shown in Fig. 2, where a symbolic analog block with a transfer characteristic $S(.)$ and a fixed scaling factor $\Delta P$ is used. Clearly, the output signal $x_o$ is related to the input $x_i$ by Eq. 1 with $P(.)$ given by Eq. 7.

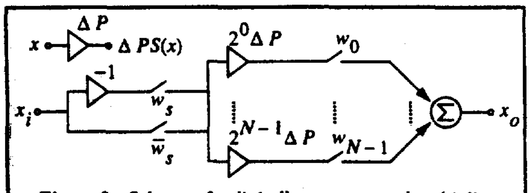

*Figure 2 - Scheme of a digitally-programmed multiplier.*

For accuracy reasons, fixed scaling factors equal to a power-of-2 multiple of $\Delta P$ are obtained by connecting several identical blocks with voltage-mode input and current-mode output in parallel. The design of these type of multipliers is straight forward after the unitary element with $\Delta P$ scaling factor has been designed. Linearity is easy to achieve because every unitary element is designed with a fixed scaling factor. Also, linearity is independent of the weight value, and the linear output signal-range scales linearly with the weight. Concerning weight accuracy, digitally-programmed multipliers are inherently robust against wafer-level process parameter variations, since the weight is given by the number of identical devices being used. Only severe variations affecting the behavior of the basic unitary block will result in wrong performance. The binary codification of the weight signal allows an easy external control of the scaling factors, since the relationship between the weight signal and the actual weight is clearly known a priory and robust against process variations. Weight values can be stored on digital memories, avoiding the time-degradation problems associated with analog memories. Finally, the same binary signals used to codify the weight value can be used to "power-off" the parts of the multiplier circuitry not being used for some particular weight value. This feature is convenient in many cases, given the high power dissipation expected from large programmable CNN systems.

The discretization of the possible weight values can be seen as a drawback of digitally-programmed multipliers. However, a fair comparison must take into account the achievable weight-accuracy of either digitally- or analog-programmed multipliers, affected by systematic and random errors (mismatch). In what follows, we use the concept of *effective* resolution, referred to the minimum relative weight increment $\Delta P/P_{\max}$ which can be achieved with a reasonable expectance of accuracy (say 90% probability of weight deviations within the range $\Delta P/2$). Assuming the general figure of about 7 to 8 bits accuracy for common, not calibrated analog circuitry [2], digitally-programmed multipliers with eight-bits weight signal will generally result in similar *effective* resolutions than many analog-programmed multipliers.

Unfortunately, there are some other relevant disadvantages. Area consumption is usually much larger than that of analog multipliers, due to the large number of unitary elements and the multiplexing circuitry required and, above this, to the large number of global control lines needed. As an example, a digitally programmable CNN with neighborhood radius of one requires a total of 19 different coefficients, nine for each of the two templates plus the offset term. If each of these coefficients is codified by an 8 bits weight-signal, the total number of global lines reaching every cell, just for weight control, is of 152. For this reason, this approach does not seem feasible.

## 4. Analog-programmed multiplier

An analog programmable multiplier can be characterized, in general, by an expression of the form,

$$x_o = h(W, x_i) \tag{8}$$

where $h(.)$ is assumed to be an approximately linear, continuous function of $x_i$, at least in some range around $x_i = 0$. We define the weight or scaling factor $P$ of the multiplier by

$$P(W) = \left.\frac{\partial h}{\partial x_i}\right|_{x_i = 0} \tag{9}$$

which is now a continuous, generally nonlinear function of the weight signal $W$. By defining

$$S(W, x_i) = \frac{h(W, x_i)}{P(W)} \tag{10}$$

we can write,

$$x_o = P(W) \cdot S(W, x_i) \tag{11}$$

It is easy to verify from the above definitions that

$$\left.\frac{\partial S}{\partial x_i}\right|_{x_i = 0} = 1 \tag{12}$$

Also, since for a given $W$ value, $S(.)$ is proportional to $h(.)$, $S(.)$ is an approximately linear, continuous function of $x_i$ in some range around $x_i = 0$. Hence, Eq. 11 is similar to the ideal formulation in Eq. 1, except that now, $S(.)$ is a function of the weight signal. This accounts for the fact that the linearity and signal range of the normalized output $x_o/P(W)$ of an analog multiplier depend, in general, on the particular value of the weight.

The design of area and power efficient analog-programmed multipliers with proper performances requires, in general, a significantly higher effort than that required for the design of digitally-programmed multipliers. In many cases, the achievable linearity of the multiplier is low for extreme weight values, unless high costs are accepted. Also, function $P(W)$ is usually a nonlinear function dependent on process parameters, difficulting the external control of the coefficients. Finally, the on-chip storage of the analog weight values requires analog memories, which lack the robustness of digital memories and present time-degradation problems.

On the other hand, there are important advantages on the use of analog-programmable multipliers. Area requirements are much lower than for digitally-programmed multipliers. In addition, the scaling factor of each multiplier implementing the same coefficient (one per cell), can be transmitted through one or two global lines.

## 5. A synergy of analog and digital programmability

Tab. 1 provides a brief comparison of the advantages and drawbacks of digital and analog programmability for the realization of general purpose CNN systems.As can be seen, most of the disadvantages of analog programmability are related to the control and storage of the weight values and their dependency on process parameters. On the other hand, digitally-programmed multipliers require large areas and an excessive number of control lines, turning them inefficient for high-density CNN implementations.

| Comparison of alternatives for Programmable CNNs | Analog Programmability | 7-8 bits Digital Programmability | Hybrid Programmability |
|---|---|---|---|
| Effective resolution | 7-8 bits | 7-8 bits | 7-8 bits |
| Area consumption | Low | Very high | Low |
| Number of internal signals | Low | Very high | Low |
| Power consumption | Variable | High | Variable |
| Process variation effects | High | Low | Low |
| Design effort | High | Low | Very high |
| External weight control | Difficult | Simple | Simple |
| Global Linearity | Difficult | Simple | Difficult |
| On-chip weight storage | Difficult | Simple | Simple |

*Table 1 - Simplified comparison of different alternatives for programmable CNNs.*

Here we propose a hybrid approach, illustrated in Fig. 3, based on the use of analog-programmed multipliers within the cells (which provides high area efficiency and low number of control lines), and digital control from the exterior of the network (which facilitates the control and on-chip storage of the weights). In this manner, the APR can be realized by a digital RAM memory.

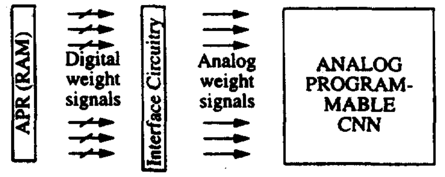

*Figure 3 - Symbolic architecture of a hybrid analog-digitally programmable CNN.*

As shown in Fig. 3, an interface circuitry is required to generate the internal analog weight-signals from their digitally coded values. This circuitry must be compound by several identical blocks, one for each programmable coefficient in the network. An uniform CNN with unitary neighborhood radius has 19 different coefficients, and hence, 19 of this interface blocks will be required in general. If the feedback and control templates are not used simultaneously, as in [3], only 10 interface blocks are required. In any case, from a system-area perspective, a reduction in the multiplier area is multiplied by the number of multipliers in every cell and by the number of cells, while the area dedicated to the interface circuitry is approximately constant.

The functionality required from the interface blocks is basically that of a nonlinear digital to analog (D/A) converter. If a linear relationship between the digital signal and the programmed weight is desired, the nonlinear characteristic of the converter must cancel out the nonlinearity of the weight function of the analog multiplier. Assume that the desired weight value is given by

$$p = \Delta P \cdot W_D \tag{13}$$

where $W_D$ and $\Delta P$ are defined by Eq. 5 and Eq. 6, respectively. If the analog multiplier being used has a weight function $p = P_A(W_A)$, we need the following transfer characteristic from the interface block

$$W_A = P_A^{-1}(p) = P_A^{-1}(\Delta P \cdot W_D) \tag{14}$$

Since the multiplier has two input signals (the weight and the input signal), its inverse function can only be defined for some fixed value of one of the inputs. For our purpose, we must set the input signal $x_i$ to a fixed reference level $x_{\text{ref}}$. The inverse function of the multiplier can be obtained using an adaptive architecture involving an analog- and a digitally-programmed multiplier, both driven by the same reference signal level $x_{\text{ref}}$, as shown in Fig. 4.

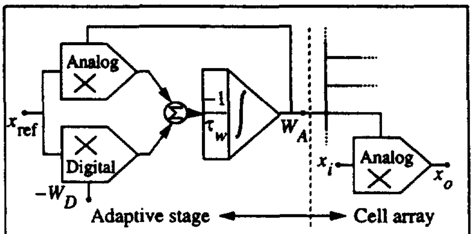

*Figure 4 - Architecture of the D/A interface circuitry required for the hybrid-control approach.*

For the analysis of this scheme, we rewrite the transfer characteristics of the analog-programmed multiplier (Eq. 11) as

$$x_o = P_A(W_A)S_A(W_A, x_i) \tag{15}$$

and that of the digitally-programmed multiplier (Eq. 1) as

$$x_o = P_D(W_D)S_D(x_i) \tag{16}$$

with $W_D$ given by Eq. 5.

The architecture in Fig. 4 can be described by following differential equation,

$$\frac{dW_A}{dt} = \frac{-1}{\tau_w}[P_A(W_A)S_A(W_A, x_{\text{ref}}) + P_D(-W_D)S_D(x_{\text{ref}})] \tag{17}$$

which defines a first order dynamical system. If function $P_A(.)$ is a monotonically increasing function of $W_A$, there is a unique equilibrium point which is locally stable, and hence, the system is globally asymptotically stable. The value of the equilibrium point is easily obtained,

$$W'_A = P_A^{-1}\left(-P_D(-W_D)\frac{S_D(x_{\text{ref}})}{S_A(W'_A, x_{\text{ref}})}\right) \tag{18}$$

If $P_D(.)$ is of the form given in Eq. 7, which is an odd function of $W_D$,

$$W'_A = P_A^{-1}\left(\Delta P \cdot W_D\frac{S_D(x_{\text{ref}})}{S_A(W'_A, x_{\text{ref}})}\right) \tag{19}$$

and if $x_{\text{ref}}$ is within the linear range of $S_D(x)$ and $S_A(W_A, x)$,

$$S_D(x_{\text{ref}}) = S_A(W'_A, x_{\text{ref}}) = x_{\text{ref}} \tag{20}$$

from where Eq. 19 results in

$$W'_A = P_A^{-1}(\Delta P \cdot W_D) \qquad (21)$$

which is the desired relationship formulated in Eq. 14.

Because this adaptation is achieved for a fixed input signal value $x_{\mathrm{ref}}$, and since analog multipliers within the network will be driven by variable signals $x_i$, multipliers offset and linearity are of extreme relevance. In particular, the obtention of Eq. 21 using the simplification in Eq. 20 must be examined carefully, attending to the real forms of $S_D$ and $S_A$, including their linear ranges and random variations.

Note that offsets in either function will result in large errors in the adapted analog weight signal, if $x_{\mathrm{ref}}$ is not much larger than average offset values (for instance measured from the standard deviation). On the other hand, large absolute values of $x_{\mathrm{ref}}$ may produce errors as well, unless the analog multiplier is highly linear. Further analysis of these error sources on the adapted analog weight signal can be carried out from Eq. 19 [4], and will not be detailed here.

Let us simply note that the digitally-programmed multiplier (a linear D/A converter) and the rest of the circuitry in the adaptive stages can be implemented using relative large areas (and hence with low errors [2]), including the analog-programmed multiplier, which can be compound of several identical blocks in parallel.

Another alternative is to use a precalibration step to cancel the offsets in the adaptive stages. This alternative, adopted in [3], requires the offsets to be approximately independent of the weight value. This property of some analog multipliers allows the use of a precalibration step to cancel the offset of the processing circuitry in the cells.

An important feature of the proposed adaptive approach is that global variation on the analog multipliers transfer characteristic have no effect on the weight values, since the adaptive scheme settles the value of the weight to that specified by the digital signal, for any analog multiplier transfer characteristic satisfying a few weak conditions.

In summary, we have proposed the use of a hybrid weight control strategy for the realization of programmable CNNs, based on automatic adaptation of the analog weight signals to the level required to set the programmable coefficients to the values specified by digital words. The weight function of the analog multiplier is required to be a monotonically increasing (or decreasing) function of the analog weight signal, and it is convenient that the random offset of the analog multipliers be independent of the weight signal. This approach merges most of the advantages of digitally-programmed and analog-programmed multipliers in the same architecture, achieving low areas and reduced number of control lines, simplifying the control and storage of the weight values, and eliminating their dependency on global process parameter variations. Last column in Tab. 1 above summarizes the advantages of the proposed hybrid control strategy. Except for the difficulty of the design, which includes global linearity and random error considerations, the best of the analog and digital programmability is exploited.

## References

[1] T. Roska and L.O. Chua: "The CNN Universal Machine: An Analogic Array Computer". *IEEE Trans. Circuits and Systems II*, Vol. 40, pp 163-173, March 1993.

[2] M.J.M Pelgrom, A.C.J. Duinmaijer and A.P.G. Welbers: "Matching Properties of MOS Transistors". *IEEE J. Solid-State Circuits*, Vol. 24, pp 1433-1440, October 1989.

[3] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez and R. Carmona: "A CNN Universal Chip in CMOS Technology". In this proceeding.

[4] S. Espejo: "*VLSI Design and Modeling of CNNs*". Ph. Dissertation, University of Sevilla, March 1994.